マルチメディア通信と分散処理　101－13
コンピュータセキュリティ　12－13
（２００１．２．２１）

IPsec処理高速化のためのSA管理方式の提案

渡辺 義則† 笠井 真理子† 中野 喜之‡

（株）日立製作所 システム開発研究所†
（株）日立システムアンドサービス‡

要旨

インターネットで利用可能な暗号通信プロトコルであるIPsecは，その標準化の進展と共に対応製品も増え始めている。そして，最近のネットワーク高速化に伴い，暗号処理による性能低下の少ない高速なIPsec対応装置のニーズが高まっている。IPsec処理の高速化には，暗号計算等を高速に行う専用ハードウェアの利用が現在は一般的である。これをさらに高速化する方法としては，専用ハードウェアを複数並列処理させる方法がある。しかし，これを実現するには，複数の専用ハードウェア間で効率良くSA(Security Association)情報を共有できることが必要である。本稿では，このような専用ハードウェアの並列処理に対応したSA管理方式を提案する。

Proposal of SA management method for high-performance IPsec processing

Yoshinori Watanabe† Mariko Kasai† Yoshiyuki Nakano‡

Systems Development Laboratory, Hitachi, Ltd.†
Hitachi Systems & Services, Ltd.‡

Abstract

IPsec is a secure communication protocol suitable for the Internet. With the progress of its standardization, many products which support IPsec are available. Recently because of using high-speed networks, a need to use high-performance IPsec products is raised. It is usual to use custom hardware for encryption to improve the performance of IPsec processing. For more improvements, the multiple custom hardware may be used in parallel. But it requires the method of sharing SA (Security Association) with the multiple custom hardware. We propose some SA management methods for the product with multiple custom hardware.

1. はじめに

企業におけるインターネット利用は近年急速に普及し，その用途は単なるWWWのアクセスやメールの利用のみならず，VPN (Virtual Private Network) 技術を利用した企業の拠点間，あるいは企業間の通信インフラへ拡大しようとしている。

インターネット上でのVPN構築に適用できる技術には種々のものが提案されており，その一つにIPsec(IPセキュリティ)と呼ばれる技術がある。

これは，インターネット技術の標準化団体で仕様が策定されている IP レベルの暗号通信プロトコルである。

標準化の進行に合わせ，最近では IPsec に対応した多くのネットワーク製品がリリースされ始め，当初は問題の多かったメーカ間での相互接続性も徐々に改善している[1]。しかし，暗号処理を伴うことから，IPsec 利用時の通信スループット低下が避けられず，ネットワークの高速化と共に，より高性能な製品のニーズも高まりつつある。

本稿では，このような IPsec 処理の高性能化のニーズに対応するため，高速に IPsec 処理を行うためのアーキテクチャと，その実現に必要な SA(Security Association)管理方式を提案する。

2. IPsec 処理高速化のアプローチ

IPsec に関連する処理をすべてソフトウェアで行った場合，暗号処理やハッシュ計算が性能ネックの主要因となる。そのため，これらの計算処理に専用ハードウェアを利用してパケットの処理性能を向上させる方式が採用され始めている。

暗号・ハッシュ計算用ハードウェアの利用形態として，例えば図 1に示すようにローカル CPU と組み合わせて IPsec パケット処理専用のコンポーネントという形でネットワーク装置に拡張する方式がある。

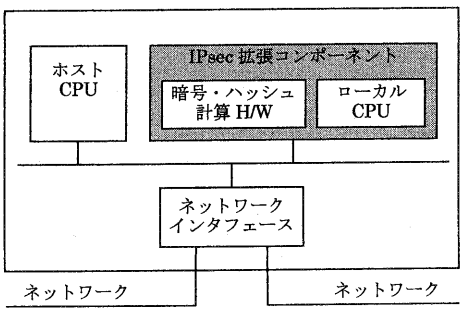

図 1 専用ハードウェアを用いた装置例

本方式では，IPsec ヘッダの処理も含めた IPsec パケット処理全体をホスト CPU から独立したコンポーネントで行うため，ホストＣＰＵに対して単純に暗号・ハッシュ計算用ハードウェアのみを拡張する場合に比べて負荷分散が可能で，ホスト CPU 側での状態管理処理も効率化できるメリットがある。ホスト CPU は，受信したパケットが IPsec 処理を必要とするものかどうかを判定し，必要なものなら IPsec 拡張コンポーネントへ転送し，不要なものならそのまま通常の IP 処理を行うのみである。

この拡張コンポーネント自体のパケットスループットは，現時点で比較的安価に利用可能な技術によっても 100Mbps 程度なら実現可能と試算している[2]。利用するネットワークとして 100Mbps 以下のものを想定するならこの程度の性能でも十分であるが，それ以上の高速なネットワークに適用する場合，さらなる高性能化が必要である。

高速化するための方法としては，拡張コンポーネントを構成する CPU や暗号・ハッシュ計算ハードウェアをより高速なものに置き換えていく方法がある。しかし，現実には CPU の放熱や消費電力といった実装上の問題，あるいはコストの問題等でこれを容易に実現できないケースもある。

そこで，今回はもう一つの高速化のアプローチとして，図 2のように拡張コンポーネントを複数用意し，複数のパケットを並列処理することで装置としてのパケットスループットを向上させる方法について，その実現上の問題点と解決方法を検討した。

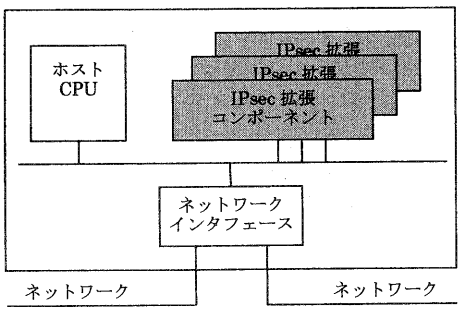

図 2 複数の拡張コンポーネントを用いた例

3. SA 管理の課題

SA (Security Association) [3]とは，IPsec 処理に必要な暗号アルゴリズムや鍵等の情報をひとまとめにした管理テーブルで，IPsec 通信を行う相手単位に管理される。SA は送信パケット用と受信パケット用が独立して管理され，IPsec による通信相手毎に少なくとも一組の送信用 SA と受信用 SA のペアが必要である。SA の主な構成要素を図 3に示す。SA の中には暗号アルゴリズムや鍵のような固定的な情報以外に，1 パケット処理毎に更新される情報も含まれる。

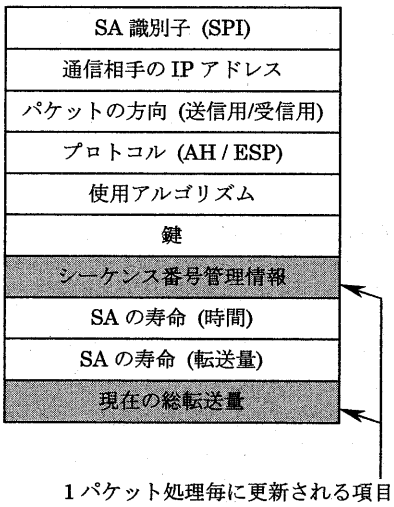

図 3　SA の内容

この SA について，IPsec 拡張コンポーネントが一つだけの場合を考えると，SA には IPsec のパケット処理中に参照する情報が多く含まれているため，拡張コンポーネント内に SA を保持する方法が SA のアクセス効率の面で有利である。

ところが，複数の IPsec 用拡張コンポーネントを持つ場合は SA の管理方法が問題となる。SA の中には 1 パケット処理毎に更新しなければならない情報が含まれるため，複数の拡張コンポーネントが SA を共有するためにはテーブルアクセスの排他制御，あるいは拡張コンポーネント間の同期制御が必要となる。

すなわち，複数の IPsec 拡張コンポーネントにより IPsec 処理の高速化を図る場合，SA 共有処理に伴うパケット処理スループットの低下が小さい SA 管理方法の実現が課題である。そして，これまで，この課題の解決方式が検討，整理されていない状態であった。

4. 並列処理対応 SA 管理方式の検討

上記の課題に対応するため，SA とそれによってパケットを実際に処理する IPsec 拡張コンポーネントの対応付け方法に着目して，いくつかの SA 管理方式を検討した。その方式の一覧を表 1に整理する。

4.1 固定割り当て方式

この方式は，SA 毎にそれを使用して IPsec 処理を行う拡張コンポーネントを一つだけ割り当てて処理の分散を図る方式である。

この方式によれば，異なる拡張コンポーネントに割り当てられている SA によるパケットの処理は並列に行われる。しかも，一つの SA は一つの拡張コンポーネントしか使用しないので，SA の同期や排他制御の問題も発生しない。

ただし，SA 毎のトラフィック量に偏りがあると，特定の拡張コンポーネントだけが高い頻度で使われ，複数の拡張コンポーネントが有効活用されないという問題が発生する。

また，本方式は，SA の拡張コンポーネントへの割り当て方法により，さらに二つの方式に分けることができる。それぞれの特徴は以下の通りである。

4.1.1 通信相手別割り当て方式

通信相手毎に使用する拡張コンポーネントを割り当てる方式である。

この方式では，通信相手が一つだけであったり，あるいは，特定の通信相手とのトラフィックだけが相対的に高いと，拡張コンポーネントの複数化による性能向上のメリットは得られない。したがって，多数の IPsec トンネルの設定を前提とする大規模セキュリティゲートウェイ装置や，多数のクライアントを収容するサ

表 1　並列処理対応 SA 管理方式の比較

方式	固定割り当て方式		動的割り当て方式
	通信相手別割り当て方式	通信方向別割り当て方式	
概要	通信相手別に使用する拡張コンポーネントを割り当てる IPsec 拡張コンポーネント #1 $SA1_{IN}$ $SA1_{OUT}$ ・・・ IPsec 拡張コンポーネント #2 $SA2_{IN}$ $SA2_{OUT}$ ・・・	パケットの方向別に使用する拡張コンポーネントを割り当てる IPsec 拡張コンポーネント #1 $SA1_{IN}$ $SA2_{IN}$ ・・・ IPsec 拡張コンポーネント #2 $SA1_{OUT}$ $SA2_{OUT}$ ・・・	全拡張コンポーネントに全 SA を割り当て，使用する拡張コンポーネントを動的に決定する IPsec 拡張コンポーネント #1 $SA1_{IN}$ $SA1_{OUT}$ ・・・ $SA2_{IN}$ $SA2_{OUT}$ IPsec 拡張コンポーネント #2 $SA1_{IN}$ $SA1_{OUT}$ ・・・ $SA2_{IN}$ $SA2_{OUT}$
実装の容易さ	SA 同期処理が不要なため容易	SA 同期処理が不要なため容易	SA 同期処理が必要なため複雑
パケット処理の並列度	通信相手毎のトラフィック量に偏りがあると並列度が上がらない	上りと下りトラフィック量に差が大きいと並列度が上がらない	トラフィックの偏りによらず並列度を高められる
適する用途	ゲートウェイ(多トンネル) サーバ	ゲートウェイ(少トンネル)	ゲートウェイ サーバ クライアント(中規模以上)

ーバにおいて有効な方式と言える。

4.1.2　通信方向別割り当て方式

パケットの種類が送信パケットか受信パケットかによって拡張コンポーネントを割り当てる方式である。この場合は，拡張コンポーネント数は基本的に二つであり，一つが送信パケット用，もう一つが受信パケット用という割り当てとなる。

この方式は，通信相手が一つであっても拡張コンポーネントの複数化による性能向上が期待できる点が，上記の通信相手別割り当て方式と異なる。ただし，使用できる拡張コンポーネント数が基本的に二つであるため，設定できるトンネル数が少ない小規模のセキュリティゲートウェイに向いている方式と言える。

一方，本方式には，送信パケットと受信パケットのトラフィック量に大きな差があると，複数の拡張コンポーネントが有効利用されないという欠点もある。そのため，一般に上りと下りのトラフィック量に大きな差があると言われている種々のサービスを提供するサーバやそのクライアントに本方式を適用しても，性能向上の効果は小さいと考える。

4.2　動的割り当て方式

本方式は，あるパケットに対しどの IPsec 拡張コンポーネントを使用するかを，パケットの IPsec 処理実行時に動的に決定する方式で，すべての SA はすべての IPsec 拡張コンポーネントから対等に参照可能とするものである。

本方式によれば，固定割り当て方式のようにトラフィックの偏りに伴う並列処理度の低下を小さくすることが可能で，今回検討した方式の中ではハードウェア資源を最も有効活用できるものである。したがって，ゲートウェイ，サーバ，クライアントのいずれにも適用可能である。

ただし，SA 参照の排他制御，同期処理が必須となる。装置全体の処理効率を落とさず，いかにこれを処理できるかが，本方式を実現する上での課題である。

本研究では，SA 内の項目と更新方法を具体的に分析し，複数の拡張コンポーネント間で SA を効率よく共有できる動的割り当て方式の実現見通しを得た。その検討結果を次章で説明する。

5. 動的割り当て方式の具体的検討

5.1 考え方

性能確保を最優先とした場合，動的割り当て方式を実現するSA管理方式に要求される条件は次の二つである。

(a) IPsecパケット処理に必要な情報は拡張コンポーネントの内のローカルCPUから高速にアクセス可能であること

(b) ホストCPU，拡張コンポーネント間でSAの更新に必要な情報のやり取りは最小限であること

たとえば，一つの方式として，すべてのSAをホストCPUが管理し，ローカルCPUが必要に応じてSA内をアクセスするためのインタフェースを用意する方式が考えられる。これはSAアクセスの排他制御は比較的容易に実現できるが，ローカルCPUからのSAアクセスがプロセッサ間通信となり，大きなオーバヘッドを伴う。したがって，上記(a)の条件を満たせない。

また，他の方式として，全SAのコピーをすべての拡張コンポーネント内に配置する方式もある。この方式だと，上記(a)の条件は満足できるが，各拡張コンポーネントが持つSA内容の同期を取るため，ホストCPUと拡張コンポーネントの間で多量のプロセッサ間通信が発生し，上記(b)の条件を満たせない。

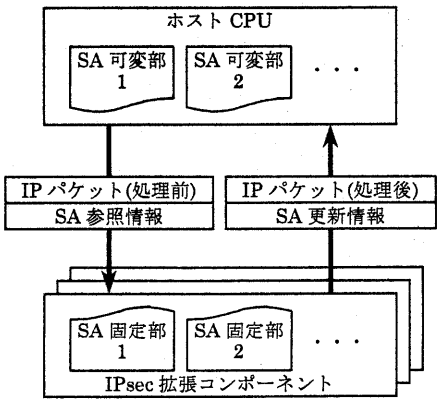

図 4 動的割り当て対応のSA管理方式

そこで，本研究では上記二つの条件を満たすために，SAを1パケット処理毎に更新しなければならない部分(以下，SA可変部と呼ぶ)と固定の部分(以下，SA固定部と呼ぶ)に分割管理する方式を考案した。SA可変部はホストCPUで管理し，SA固定部はそのコピーを各拡張コンポーネントが保持する。そして，SA可変部の参照，更新に必要な情報は，ホストCPUと拡張コンポーネント間でやり取りするIPパケットに付加する。この様子を図4に示す。

5.2 SA参照・更新方法

図3に示した項目のうち，SA可変部に格納される情報は「SA識別子」，「シーケンス番号管理情報」，「現在の総転送量」である。SA識別子は固定的な情報であるが，テーブルの管理上必要である。

ここでは，これらの情報の参照と更新が図4に示したような形態の情報交換のみで可能であることをSA項目毎に示す。

5.2.1 シーケンス番号管理情報

シーケンス番号とは，IPsecパケットに付加されるリプレイ攻撃[3]チェック用の32ビットの番号である。

送信側は，1パケット毎にシーケンス番号を1ずつ増やし，受信側は一度受信したシーケンス番号のパケットは破棄することでリプレイ攻撃を防ぐ仕掛になっている[4][5]。

IPの場合，パケットの欠落や順序入れ替えの可能性があるので，受信用のシーケンス番号管理情報内には，指定のウィンドウサイズ分だけ過去に受信したパケットのシーケンス番号を保持するテーブルを持つ。また，この情報によるパケットのチェックはIPsec処理前に行わなければならないが，この情報の更新はIPsec処理後でIPsec認証が通った場合でないと行うことができない。

以上から検討すると，本情報の更新に関しホストCPUと拡張コンポーネント間でやり取りされる情報は表2のようになる。

5.2.2 現在の総転送量

この情報は，暗号鍵等の寿命を，それを含むSA を使ったトラフィックの総量により管理する場合に必要となる情報である。

これは，拡張コンポーネントによるIPsec処理に成功したとき，実際にそのSA内の鍵を使って暗号，復号，認証計算を行ったデータ量を積算すれば良い。したがって，本情報の更新に関しホストCPUと拡張コンポーネント間でやり取りされる情報は表3のようになる。

表2　シーケンス番号管理情報の参照と更新

情報種別		内容
送信	参照情報	シーケンス番号(4バイト) (本情報送信後，ホストCPUはシーケンス番号更新)
	更新情報	なし
受信	参照情報	なし (ホストCPUはシーケンス番号を事前チェック)
	更新情報	シーケンス番号(4バイト) (本情報送信後，ホストCPUはシーケンス番号管理情報更新)

表3　現在の総転送量の参照と更新

情報種別		内容
送信	参照情報	なし
	更新情報	処理バイト数(4バイト) (ホストCPUはこれを総転送量に積算)
受信	参照情報	なし
	更新情報	処理バイト数(4バイト) (ホストCPUはこれを総転送量に積算)

以上の議論から，図4に示したSA管理方式を実現するにあたり，IPパケットに付加するSA参照情報，SA更新情報は，多くてもそれぞれ十数バイト程度で済むものと考えられる。IPパケットのサイズが通常は40〜1500バイト程度であることを考えると，この付加情報がホストCPUと拡張コンポーネントの通信負荷になるとは考えにくく，暗号鍵等の情報は拡張コンポーネント内に置かれることから，先に示したSA管理に要求される二つの条件を満足する方式であると考える。

6. まとめと今後の課題

複数のIPsec処理用拡張コンポーネントによるIPsec高速化方式に適用するSA管理方式を検討し，実装が容易でトラフィックが均一であればハードウェア資源を有効利用できる固定割り当て方式と，実装は複雑であるがトラフィックに偏りがあってもハードウェア資源を有効利用できる動的割り当て方式を示した。

ただし，動的割り当て方式では，固定割り当て方式の場合に拡張コンポーネントが担当していた処理の一部をホストCPU側で処理することになるため，これが原因で性能が期待するほど向上しない可能性もある。今後は，プロトタイプ作成とその性能評価による本提案方式の有効性検証が必要である。

また，拡張コンポーネントの数を増やした場合，それが接続されるバスの帯域が性能のネックになる。このようなケースでの性能評価を行っていくことも今後の課題である。

参考文献

[1] 渡辺義則，大浦哲生：IPsecの相互接続性に関する現状と課題：情報処理学会研究報告 99-CSEC-6 pp.31-36, Jul. 1999

[2] 笠井真理子，渡辺義則，中野喜之：IPsec処理の高速化方式の検討：情報処理学会研究報告 01-CSEC-12, Feb. 2001

[3] RFC2401 "Security Architecture for the Internet Protocol", http://www.ietf.org/rfc.html

[4] RFC2402 "IP Authentication Header", http://www.ietf.org/rfc.html

[5] RFC2406 "IP Encapsulating Security Payload (ESP)", http://www.ietf.org/rfc.html